

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Achtermann et al.** §
Serial No. **09/438,436** § Group Art Unit: **2157**
Filed: **November 12, 1999** § Examiner: **Todd, Gregory G.**
For: **Apparatus for Connection** §
Management and the Method §
Therefor §

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

35525
PATENT TRADEMARK OFFICE
CUSTOMER NUMBER

APPEAL BRIEF (37 C.F.R. 41.37)

This brief is in furtherance of the Reinstatement of Appeal, filed in this case on 10/02/07.

No fees are believed to be required. If, however, any fees are required, I authorize the Commissioner to charge these fees which may be required to IBM Corporation Deposit Account No. 09-0447.

A one-month extension of time is believed to be necessary. I authorize the Commissioner to charge the one-month extension fee of \$120.00 to Yee & Associates, P.C. Deposit Account No. 50-3157. No additional extension of time is believed to be necessary. If, however, an additional extension of time is required, the extension is requested, and I authorize the Commissioner to charge any fees for this extension to Yee & Associates, P.C. Deposit Account No. 50-3157.

REAL PARTY IN INTEREST

The real party in interest in this appeal is the following party: International Business Machines Corporation of Armonk, New York.

RELATED APPEALS AND INTERFERENCES

With respect to other appeals or interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no such appeals or interferences.

STATUS OF CLAIMS

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

Claims in the application are: 1-33

B. STATUS OF ALL THE CLAIMS IN APPLICATION

1. Claims canceled: 3, 5, 14, 16, 25 and 27
2. Claims withdrawn from consideration but not canceled: none
3. Claims pending: 1, 2, 4, 6-13, 15, 17-24, 26 and 28-33
4. Claims allowed: none
5. Claims rejected: 1, 2, 4, 6-13, 15, 17-24, 26 and 28-33
6. Claims objected to: none

C. CLAIMS ON APPEAL

The claims on appeal are: 1, 2, 4, 6-13, 15, 17-24, 26 and 28-33

STATUS OF AMENDMENTS

An amendment pursuant to 37 C.F.R. 41.33(a) and MPEP 1206 was filed by Appellants on November 6, 2007. In an advisory action dated November 21, 2007, the Examiner indicated such amendment would be entered.

SUMMARY OF CLAIMED SUBJECT MATTER

A connection scheduling method that is operable in a node of a networked data processing system comprising a plurality of nodes. A determination is made as to whether a job is available for scheduling. In response to this step of determining if the job is available, a determination is made as to whether a session is available, where the session is included in a pool of sessions, with the pool of sessions having a preselected one of a set of priority levels corresponding to a priority level of the job and the session effects an execution of the job. *A network connection to a target system is created for the execution of the job, where the target system is another node of the networked data processing system.* If the session is available, the session is launched to effect the execution of the job. An error handling thread is launched in response to an error condition, with the error handling thread releasing the session. Thus, per the claimed features as further described below, a bulk data distribution mechanism is provided that can transfer large amounts of data between networked connected subsystems while maintaining scalability.

A. CLAIM 1 - INDEPENDENT

Independent claim 1:

The invention embodiment claimed in Appellants' independent claim 1 provides a connection scheduling method (Appellants' specification, page 8, lines 3-12). The claimed connection scheduling method embodiment comprises determining if a job is available for scheduling (Appellants' specification, page 14, lines 1-14); determining if a session is available in response to said step of determining if said job is available (Appellants' specification, page 15, lines 1-3), wherein said session is included in a pool of sessions (Appellants' specification, page 15, lines 3-12), said pool of sessions having a preselected one of a set of priority levels corresponding to a priority level of said job and wherein said session effects an execution of said job (Appellants' specification, page 15, line 13 to page 16, line 2); launching said session to effect said execution of said job, if said session is available (Appellants' specification, page 19, lines 7-9

9); and launching an error handling thread in response to an error condition, said error handling thread releasing said session (Appellants' specification, page 20, lines 4-6).

Dependent claim 4:

The invention embodiment claimed in Appellants' independent claim 4 comprises determining if said connection is an existing connection (Appellants' specification, page 18, lines 4-6), and wherein said step of creating said connection is performed if said connection is not an existing connection (Appellants' specification, page 18, lines 14-15).

Dependent claim 6:

The invention embodiment claimed in Appellants' independent claim 6 comprises changing value of a job state from a first value to a second value in response to said launching of said error handling thread (Appellants' specification, page 20, lines 11-12).

Dependent claim 7:

The invention embodiment claimed in Appellants' independent claim 7 provides for said first value signals that said job is available for scheduling (Appellants' specification, page 17, line 13, to page 18, line 3).

Dependent claim 8:

The invention embodiment claimed in Appellants' independent claim 8 comprises retrying said steps of determining if a job is available for scheduling (Appellants' specification, page 19, lines 11-13), determining if a session is available (Appellants' specification, page 19, line 13), and launching said session, in response to an error condition (Appellants' specification, page 19, lines 13-16).

Dependent claim 9:

The invention embodiment claimed in Appellants' independent claim 9 comprises retrying is repeated until a predetermined time interval has elapsed (Appellants' specification, page 21, lines 6-8).

B. CLAIM 12 - INDEPENDENT

Independent claim 12:

The invention embodiment claimed in Appellants' independent claim 12 provides a data processing system for connection scheduling (Appellants' specification, page 8, lines 3-12). The claimed data processing system for connection scheduling provides circuitry operable for determining if a job is available for scheduling (Appellants' specification, page 14, lines 1-14) provides circuitry operable for determining if a session is available, in response to said circuitry operable for determining if said job is available (Appellants' specification, page 15, lines 1-3), wherein said session is included in a pool of sessions (Appellants' specification, page 15, lines 3-12), said pool of sessions having a preselected one of a set of priority levels corresponding to a priority level of said job and wherein said session effects an execution of said job (Appellants' specification, page 15, line 13 to page 16, line 2); provides circuitry operable for launching said session to effect said execution of said job, if said session is available (Appellants' specification, page 19, lines 7-9); and provides circuitry operable for launching an error handling thread in response to an error condition, said error handling thread releasing said session (Appellants' specification, page 20, lines 4-6).

The system recited in claim 12, as well as dependent claims 13-15 and 19-22, may be a bus system comprised of system bus **212**; I/O adapter **218**; communication adapter **234**, memory comprised of read only memory **216** and random access memory **214**, and central processing unit **210** performing the steps described in the specification at page 13, line 7, to page 23, line 3, or equivalent.

Dependent claim 15:

The invention embodiment claimed in Appellants' independent claim 15 provides circuitry operable for determining if said connection is an existing connection (Appellants' specification, page 18, lines 4-6), and wherein said circuitry operable for creating said connection is operated if said connection is not an existing connection (Appellants' specification, page 18, lines 14-15).

Dependent claim 17:

The invention embodiment claimed in Appellants' independent claim 17 provides for circuitry operable for changing value of a job state from a first value to a second value in response to said launching of said error handling thread (Appellants' specification, page 20, lines 11-12).

Dependent claim 18:

The invention embodiment claimed in Appellants' independent claim 18 provides for said first value signals that said job is available for scheduling (Appellants' specification, page 17, line 13, to page 18, line 3).

Dependent claim 19:

The invention embodiment claimed in Appellants' independent claim 19 provides circuitry operable for retrying said steps of determining if a job is available for scheduling (Appellants' specification, page 19, lines 11-13), determining if a session is available (Appellants' specification, page 19, line 13), and launching said session, in response to an error condition (Appellants' specification, page 19, lines 13-16).

Dependent claim 20:

The invention embodiment claimed in Appellants' independent claim 20 provides circuitry operable for retrying is operated until a predetermined time interval has elapsed (Appellants' specification, page 21, lines 6-8).

C. CLAIM 23 - INDEPENDENT

Independent claim 23:

The invention embodiment claimed in Appellants' independent claim 23 provides a computer program product embodied in a machine readable storage medium, the program product for job scheduling comprising instructions (Appellants' specification, page 8, lines 3-12). The claimed computer program product for job scheduling provides instructions for determining if a job is available for scheduling (Appellants' specification, page 14, lines 1-14); instructions for determining if a session is available in response to instructions for determining if said job is available (Appellants' specification, page 15, lines 1-3) , wherein said session is included in a pool of sessions (Appellants' specification, page 15, lines 3-12), said pool of sessions having a preselected one of a set of priority levels corresponding to a priority level of said job and wherein said session effects an execution of said job (Appellants' specification, page 15, line 13 to page 16, line 2); instructions for launching said session to effect said execution of said job, if said session is available (Appellants' specification, page 19, line 7-9); and instructions for launching an error handling thread in response to an error condition, said error handling thread releasing said session (Appellants' specification, page 20, lines 4-6).

A person having ordinary skill in the art would be able to derive computer instructions on a computer readable medium as recited in claim 23, as well as dependent claims 24-26 and 30-33, given **Figure 2** and the corresponding description at page 13, line 7, to page 23, line 3, without undue experimentation.

Dependent claim 26:

The invention embodiment claimed in Appellants' independent claim 26 provides instructions for determining if said connection is an existing connection (Appellants' specification, page 18, lines 4-6), and wherein said instructions for creating said connection are performed if said connection is not an existing connection (Appellants' specification, page 18, lines 14-15).

Dependent claim 28:

The invention embodiment claimed in Appellants' independent claim 28 provides instructions for changing value of a job state from a first value to a second value in response to said launching of said error handling thread (Appellants' specification, page 20, lines 11-12).

Dependent claim 29:

The invention embodiment claimed in Appellants' independent claim 29 provides for said first value signals that said job is available for scheduling (Appellants' specification, page 17, line 13, to page 18, line 3).

Dependent claim 30:

The invention embodiment claimed in Appellants' independent claim 30 provides instructions for retrying said steps of determining if a job is available for scheduling (Appellants' specification, page 19, lines 11-13), determining if a session is available (Appellants' specification, page 19, line 13), and launching said session, in response to an error condition (Appellants' specification, page 19, lines 13-16).

Dependent claim 31:

The invention embodiment claimed in Appellants' independent claim 31 provides instructions for retrying are repeated until a predetermined time interval has elapsed (Appellants' specification, page 21, lines 6-8).

GROUND OF REJECTION TO BE REVIEWED ON APPEAL

The grounds of rejection to review on appeal are as follows:

1. Whether Claims 4, 15 and 26 are indefinite under 35 U.S.C. § 112, 2nd paragraph;
2. Whether Claims 1-2, 12-13 and 23-24 are obvious over Zolnowsky (US Patent No. 6,779,182, hereinafter “Zolnowsky”) in view of Kumpf et al. (US Patent No. 6,289,371, hereinafter “Kumpf”) under 35 U.S.C. § 103;
3. Whether Claims 4, 15 and 26 are obvious over Zolnowsky (US Patent No. 6,779,182) in view of Kumpf et al. (US Patent No. 6,289,371) and further in view of Sayan et al. (US Patent No. 6,477,569, hereinafter “Sayan”) under 35 U.S.C. § 103;
4. Whether Claims 6-9, 17-20 and 28-31 are obvious over Zolnowsky (US Patent No. 6,779,182 in view of Kumpf et al. (US Patent No. 6,289,371) and further in view of Northrup (US Patent No. 6,671,713, hereinafter “Northrup”) under 35 U.S.C. § 103; and
5. Whether Claims 10-11, 21-22 and 32-33 are obvious over Zolnowsky (US Patent No. 6,779,182 in view of Kumpf et al. (US Patent No. 6,289,371) and further in view of Northrup (US Patent No. 6,671,713) and Rangarajan et al. (US Patent No. 6,260,077, hereinafter “Rangarajan”) under 35 U.S.C. § 103.

ARGUMENT

A. GROUND OF REJECTION 1 (Claims 4, 15 and 26)

Claims 4, 15 and 26 stand rejected under 35 U.S.C. § 112, 2nd paragraph as being indefinite.

A.1. Claims 4, 15 and 26

In a previous amendment filed by Appellants on April 11, 2007, the features of dependent Claims 3, 14 and 25 were added to independent Claims 1, 12 and 23, respectively. However, dependent Claims 4, 15 and 26 were not amended to correctly reference a pending claim. An amendment pursuant to 37 C.F.R. 41.33(a) and MPEP 1206 was filed by Appellants on November 6, 2007 to correct the current antecedent basis issue with respect to these Claims 4, 15 and 26. In an Advisory Action dated November 21, 2007, the Examiner indicated such amendment overcomes the present 35 U.S.C. § 112, 2nd paragraph rejection and would be entered. It is urged that this rejection of Claims 4, 15 and 26 under 35 U.S.C. § 112, 2nd paragraph is now moot.

B. GROUND OF REJECTION 2 (Claims 1-2, 12-13 and 23-24)

Claims 1-2, 12-13 and 23-24 stand rejected under 35 U.S.C. § 103 as being unpatentable over Zolnowsky in view of Kumpf et al.

B.1. Claims 1-2, 12-13 and 23-24

The question of obviousness is resolved on the basis of underlying factual determinations including (1) the scope and content of the prior art, (2) any differences between the claimed subject matter and the prior art, (3) the level of skill in the art, and (4) where in evidence, so-called secondary considerations. *Graham v. John Deere Co.*, 383 U.S. 1, 17-18 (1966); *see also KSR Int'l Co. v. Teleflex Inc.*, 127 S. Ct. 1727, 1734 (2007) (“While the sequence of these questions might be reordered in any particular case, the [Graham] factors continue to define the inquiry that controls.”). In *KSR*, the Supreme Court also reaffirmed principals based on its precedent that “[t]he combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results”. *Id.* The operative question in this “functional approach” is thus “whether the improvement is more than the predictable use of

prior art elements according to their established functions.” *Id.* The court went on to state that “a court must ask whether the improvement is more than the predictable use of prior art elements according to their established function.” *Id.* at 1740.

With respect to Claim 1, such claim recites “creating a network connection to a target system for said execution of said job, *wherein said target system is another node of the networked data processing system*” (emphasis added by Appellants). In rejecting this claimed feature, the Examiner cites Zolnowsky’s teaching at col. 10, lines 21-42 as teaching the claimed feature of creating a connection to a target system for said execution of said job. Appellants urge that, instead, this cited passage describes ‘selecting a processor’ and does not teach or otherwise suggest any creation of a connection to a target system. For example, as stated by Zolnowsky (col. 10, lines 21-42):

FIG. 10 illustrates the processor selection step 905 of FIG. 9 in greater detail. At step 1001, “best processor” is set to be the processor the thread last ran on. Then each processor in the system determines if its current thread is of lower priority than the thread being inserted at step 1002. If its current thread is of higher priority, then it proceeds to step 1005. Otherwise, a determination is made at step 1003 whether the processor’s current thread is of lower priority than the thread on “best processor.” If the current thread is of lower priority, then at step 1004, “best processor” is set equal to current processor. At step 1005, “best processor” is selected as the target processor to schedule the thread being inserted.

Turning back to FIG. 9, after a target processor is selected to run the thread, a determination is made, at decision block 907, as to whether the thread has a higher priority than the last thread the processor committed to. If the thread has a higher priority, the processor is notified at step 908 by setting some local variables, for example `cpu_runrun` and `cpu_kprunrun` variables. If the thread does not have a higher priority at step 907, the scheduling processor continues to step 909.

As can be seen, this cited passage describes that a best processor is selected as a target processor to schedule a thread. Selection of a target processor does not teach or otherwise suggest creation of a connection to a target system, as expressly recited in Claim 1.

In addition, Claim 1 recites features that further emphasize the networked aspect of the present invention – and in particular that the connection scheduling technique is with respect to the *scheduling of network connections*. In contrast, the cited Zolnowsky reference contemplates a scheduling technique in a non-networked environment – i.e. in a multiprocessor data processing system having a single operating system executing therein (Zolnowsky col. 1, lines 19-23; col. 2, lines 47-56; col. 3, lines 65-67). Because Zolnowsky’s thread scheduling technique *operates solely within* such a multiprocessor data processing system, and in particular is directed to the *scheduling of internal threads* to meet real-time application requirements (Zolnowsky col. 6, lines 32-65), there would have been no reason or other motivation to create any type of network connection to another node of a networked data processing system as a part of a connection scheduling methodology, as claimed.

In fact, the Zolnowsky internal thread scheduling system could not be used to schedule connections to another node in a multi-node, networked data processing system due to its requirement of a global dispatch queue that is directly accessible by all processors so that a given processor’s dispatch can select a thread for execution from the global dispatch queue, its own queue, or another processor’s queue (Zolnowsky col. 4, lines 1-20). This Zolnowsky architecture is specifically designed to mitigate internal race conditions (Zolnowsky col. 4, lines 21-27), and could not be used in a multi-node, networked environment as there is no global dispatch queue in such a multi-node, networked environment – and such global dispatch queue is expressly required by Zolnowsky’s system to eliminate the internal race conditions for which the Zolnowsky system was specially designed for.

Thus, it is urged that Claim 1 is not obvious in view of the cited references as there are missing claimed features that are not taught, suggested or logical flow from the teachings of the cited references – and the differences between what is claimed and what is described by the cited references are substantial and are much more than the predictable use of prior art elements according to their established functions.

Therefore, as described above, Claim 1 (and similarly for Claims 2, 12-13 and 23-24) has been erroneously rejected under 35 U.S.C. § 103.

C. GROUND OF REJECTION 3 (Claims 4, 15 and 26)

Claims 4, 15 and 26 stand rejected under 35 U.S.C. § 103 as being unpatentable over Zolnowsky in view of Kumpf and further in view of Sayan.

C.1. Claims 4, 15 and 26

With respect to Claim 4 (and similarly for Claims 15 and 26), Appellants initially show error in the rejection of such claim for reasons given above with respect to Claim 1 (of which Claim 4 depends upon), and urge that the additional cited reference to Sayan does not overcome the teaching/suggestion deficiency identified above regarding the network connection scheduling methodology recited in Claim 1.

Further with respect to Claim 4, such claim recites “determining if said network connection is an existing network connection, and wherein said step of creating said network connection is performed *if* said network connection is *not* an existing network connection, and wherein said session is launched using said existing network connection if said network connection is an existing network connection such that said existing network connection supports multiple logical sessions” (emphasis added). As can be seen, per the features of Claim 4, a determination is made as to whether the network connection is an existing network connection, and the network connection creation step is conditionally performed based on this determination – specifically, the network connection creation step is conditionally performed *if* the network connection is *not* an existing network connection. In rejecting this aspect of Claim 4, the Examiner acknowledges that neither of the cited references to Zolnowsky or Kumpf teaches such conditional network connection creation. However, the Examiner alleges that Sayan teaches such conditional network connection creation at Sayan col. 2, lines 26-38 and col. 10, lines 52-67. Appellants urge error in such assertion. For example, Sayan describes at the cited col. 2 passage that multiple instantiations of network services are installed on network servers. This cited passage also describes an ability for a given such instantiation to simultaneously service multiple client terminals. This cited passage also describes that a client terminal may simultaneously establish a logical connection with multiple services – either on one or multiple servers. This cited passage does not describe any type of conditional network connection creation.

As to the cited Sayan passage at col. 10, such passage similarly fails to describe any type of conditional network connection creation, where a network connection is conditionally created *if* said network connection is *not* an existing network connection. Instead, this passage describes conditional processing with respect to *agent availability*, and not with respect to the status of a network connection. For example, upon receipt of a request for a connection, the pool master selects an idle agent and gives that agent the connection request to process. This step is not conditioned upon whether there is an existing network connection. This cited passage further states that if there are *no idle agents* available, the pool master will not create a new agent but instead find an existing agent with resources available and pass the connection request to such agent. Thus, while this is a conditional processing step, the condition for which such processing is conditioned upon is *whether or not there is an idle agent or not* – and not whether or not a network connection is an existing network connection. Quite simply, idle agents (as per the teachings of the cited reference) are not network connections to other nodes in a multi-node, networked data processing system (as claimed).

Even if one were to improperly construe agent availability, as described by Sayan, to be equivalent to whether a network connection exists, as claimed, even then the Sayan reference describes exactly the opposite scenario from what is claimed. For example – and again assuming arguendo that agent available is equivalent to network connection existence – the cited reference describes that *if there are no existing connections* (no idle agents), another connection (*existing agent*) is used to satisfy the connection request. Claim 4 recites just the opposite scenario, whereby if there is *no connection*, a *new one is created*. Thus, even when improperly interpreting Sayan's agent to being equivalent to the claimed connection, such improper interpretation describes using an existing resource (existing agent) when no requested resource (connection/idle agent) exists.

Quite simply, none of the cited references describe conditional processing with respect to the existence, or non-existence, of a network connection. Therefore, a proper prima facie case of obviousness has not been established with respect to Claim 4, as none of the cited references teach, suggest, or logically flow in their established function to provide a conditional creation of a network connection that is performed if the network connection is not an existing network

connection¹. Thus, it is further shown that Claim 4 has been erroneously rejected, as a proper prima facie showing of obviousness has not been established².

D. GROUND OF REJECTION 4 (Claims 6-9, 17-20 and 28-31)

Claims 6-9, 17-20 and 28-31 stand rejected under 35 U.S.C. § 103 as being unpatentable over Zolnowsky in view of Kumpf and further in view of Northrup.

D.1. Claims 6-7, 17-18 and 28-29

With respect to Claims 6-7 (and similarly for Claims 17-18 and 28-29), Appellants initially show error in the rejection of such claims for reasons given above with respect to Claim 1 (of which Claims 6-7 depend upon), and urge that the additional cited reference to Northrup does not overcome the teaching/suggestion deficiency identified above regarding the network connection scheduling methodology recited in Claim 1.

D.2. Claims 8-9, 19-20 and 30-31

With respect to Claims 8-9 (and similarly for Claims 19-20 and 30-31), Appellants initially show error in the rejection of such claims for reasons given above with respect to Claim 1 (of which Claims 8-9 depend upon), and urge that the additional cited reference to Northrup does not overcome the teaching/suggestion deficiency identified above regarding the network connection scheduling methodology recited in Claim 1.

Further with respect to Claim 8 (and dependent Claim 9), such claim recites “The method of claim 1 further comprising the step of retrying said steps of determining if a job is available for scheduling, determining if a session is available, and launching said session, in response to an error condition”. The Examiner states that the step of retrying is taught by Zolnowsky col. 8, lines 11-17. Appellants show that there, Zolnowsky states:

¹ To establish prima facie obviousness of a claimed invention, all of the claim limitations must be taught or suggested by the prior art. MPEP 2143.03. *See also, In re Royka*, 490 F.2d 580 (C.C.P.A. 1974).

² If the examiner fails to establish a prima facie case, the rejection is improper and will be overturned. *In re Fine*, 837 F.2d 1071, 1074, 5 USPQ2d 1596, 1598 (Fed. Cir. 1988).

“However, any other suitable synchronization algorithm can be used in alternate embodiments of the present invention. Since the priority variables to be examined are atomic variables that are maintained in each dispatch queue, any scheduling errors caused by selecting a wrong queue will be caught in the a verification step. However, a schedule lock is required to take a thread from a selected queue.”

As can be seen, this cited passage does allude to errors, but that is the extent of any similarities between the teachings of this cited passage and the features recited in Claim 8. Specifically, this passage merely states that scheduling errors ‘will be caught’ in the verification step. This passage does not teach or otherwise suggest that three specifically enumerated steps of (i) determining if a job is available, (ii) determining if a session is available, and (iii) launching the session are retried in response to an error condition. In fact, this passage does not describe anything being done in response to an error condition. Rather, this passage merely states that an error is ‘caught’. Thus, it is further urged that Claim 8 (and dependent Claim 9) has been erroneously rejected, as a proper prima facie showing of obviousness has not been established³.

E. GROUND OF REJECTION 5 (Claims 10-11, 21-22 and 32-33)

Claims 10-11, 21-22 and 32-33 stand rejected under 35 U.S.C. § 103 as being unpatentable over Zolnowsky in view of Kumpf and Northrup and further in view of Rangarajan.

E.1. Claims 10, 21 and 32

With respect to Claim 10 (and similarly for Claims 21 and 32), Appellants show error in the rejection of such claim for reasons given above with respect to Claim 1 (of which Claim 10 ultimately depends upon), and urge that the additional cited references to Northrup and Rangarajan do not overcome the teaching/suggestion deficiency identified above regarding the network connection scheduling methodology recited in Claim 1.

³ MPEP 2143.03, *In re Fine*, *supra*.

E.2. Claims 11, 22 and 33

With respect to Claim 11 (and similarly for Claims 22 and 33), Appellants show error in the rejection of such claim for reasons given above with respect to Claim 1 (of which Claim 11 ultimately depends upon), and urge that the additional cited references to Northrup and Rangarajan do not overcome the teaching/suggestion deficiency identified above regarding the *network connection scheduling* methodology recited in Claim 1.

Further with respect to Claim 11 (and similarly for Claims 22 and 33), Appellants show error in the rejection of such claim for reasons given above with respect to Claim 8 (of which Claim 11 depends upon), and urge that the additional cited references to Northrup and Rangarajan do not overcome the teaching/suggestion deficiency identified above regarding the *retrying of job availability determination, session availability determination, and session launching* that is recited in Claim 8.

Further with respect to Claim 11, such claim recites “wherein said steps of determining if a job is available for scheduling, determining if a session is available, and launching said session are performed in response to an invoking of said callback method by said target system”. As can be seen, the three retry steps recited in Claim 8 are performed in response to a target system invoking a callback method – with such callback method being registered in response to a time interval expiration (per Claim 10). In rejecting Claim 11, the Examiner states:

“Zolnowsky, Kumpf and Northrup fail to explicitly disclose the steps of determining if a job is available for scheduling, determining if a session is available, and launching said session being performed in response to an invoking of a callback method by a target system, the target system for execution of said job. However, the use and advantages for a target system responding to an elapsed time expiration is well known to one skilled in the art at the time the invention was made as evidenced by the teachings of Rangarajan”

Appellants urge that even assuming arguendo that a target system responding to an elapsed time expiration is well known, such well-known assertion does not establish the specific claimed features recited in Claim 11 of “wherein said steps of determining if a job is available for

scheduling, determining if a session is available, and launching said session *are performed in response to an invoking of said callback method* by said target system”. Even if it were well known that a target system responds to an elapsed time expiration, such well known assertion does not establish a specific teaching or suggestion that particular actions/steps are performed (e.g., ‘determining’, ‘launching’) *in response to* such an elapsed time expiration. Thus, a proper prima facie showing of obviousness has not been established by the Examiner, and thus Claim 11 has been erroneously rejected under 35 U.S.C. § 103.

Still further, the cited Rangarajan reference does not evidence a target system responding to an elapsed time expiration, as alleged by the Examiner. The Examiner cites Rangarajan’s teaching at Abstract and col. 17, lines 13-39 to support such assertion. It is urged that Rangarajan’s Abstract describes that programmed-method invocations performed on logical objects at a client are communicated to the server object. There is no mention of any timer or timer-associated operations. As to the cited passage at col. 17, there Rangarajan describes particular callback methods – an alarmRecordCreated, an alarmRecordDeleted, and an alarmRecordModified call-back method. Each of these three callback methods are invoked *when a record is changed* in the client, including a new record being created, and existing record being deleted or an existing record being modified. It is urged that the creation, deletion or modification of a record, as described by Rangarajan, is not done *in response to an elapsed time expiration*, as required by Claim 11 in combination with Claim 10.

It is further shown that even if the Rangarajan call-back method occurred in response to expiration of a time interval (which it does not), there is still no teaching in the combination of references of *retrying the three specifically enumerated steps recited in Claim 8 in response to such a call-back*. The parameters returned by Rangarajan in the described call-back methods are all related to record identifiers – specifically (i) an object that provides access to the new record (col. 17, lines 26-28), (ii) an object that identifies a deleted alarm record (col. 17, 33-34), and (iii) an object that provides access to the modified record (col. 17, lines 37-38). None of the parameters that are returned by these routines provide any type of indication that a retry operation is needed.

Further still, none of the Rangarajan call-back methods occur in response to any type of error condition, so there would be no reason for any recipient of such call-back methods to perform any type of error recovery such as the claimed retry of the three specifically enumerated

steps recited in Claim 8. Thus, the only reason for asserting a teaching of an error retry condition in response to a call-back method must be coming from Applicants' own Specification and claims, which is impermissible hindsight analysis⁴. Thus, it is further urged that Claim 11 has been erroneously rejected using impermissible hindsight analysis.

Appellants have thus shown numerous and substantial error in the final rejection of all pending claims, and therefore respectfully requests that the Board reverse the final rejection of all such claims in this third appeal of this case.

/Wayne P. Bailey/
Wayne P. Bailey
Reg. No. 34,289
YEE & ASSOCIATES, P.C.
PO Box 802333
Dallas, TX 75380
(972) 385-8777

⁴ It is error to reconstruct the patentee's claimed invention from the prior art by using the patentee's claims as a "blueprint". When prior art references require selective combination to render obvious a subsequent invention, there must be some reason for the combination other than the hindsight obtained from the invention itself. *Interconnect Planning Corp. v. Feil*, 774 F.2d 1132, 227 USPQ 543 (Fed. Cir. 1985). In 1983, the late Judge Howard Markey made the following observation in *W.L. Gore & Associates Inc. v. Garlock, Inc.*, 721 F.2d 1540, 220 USPQ 303 (Fed. Cir. 1983), which states the basic interest protected by this test—improper hindsight analysis of prior art:

To imbue one of ordinary skill in the art with knowledge of the invention in suit, when no prior art reference or references of record convey or suggest that knowledge, is to fall victim to the insidious effect of a hindsight syndrome wherein that which only the inventor taught is used against its teacher.

CLAIMS APPENDIX

The text of the claims involved in the appeal are:

1. A connection scheduling method, operable in a node of a networked data processing system comprising a plurality of nodes, comprising the steps of:
 - determining if a job is available for scheduling;
 - determining, in response to said step of determining if said job is available, if a session is available, wherein said session is included in a pool of sessions, said pool of sessions having a preselected one of a set of priority levels corresponding to a priority level of said job and wherein said session effects an execution of said job;
 - creating a network connection to a target system for said execution of said job, wherein said target system is another node of the networked data processing system;
 - launching said session to effect said execution of said job, if said session is available; and
 - launching an error handling thread in response to an error condition, said error handling thread releasing said session.
2. The method of claim 1 wherein said session comprises a thread.
4. The method of claim 1 further comprising the step of determining if said network connection is an existing network connection, and wherein said step of creating said network connection is performed if said network connection is not an existing network connection, and wherein said session is launched using said existing network connection if said network

connection is an existing network connection such that said existing network connection supports multiple logical sessions.

6. The method of claim 1 further comprising the step of changing value of a job state from a first value to a second value in response to said launching of said error handling thread.

7. The method of claim 6 wherein said first value signals that said job is available for scheduling.

8. The method of claim 1 further comprising the step of retrying said steps of determining if a job is available for scheduling, determining if a session is available, and launching said session, in response to an error condition.

9. The method of claim 8 wherein said step of retrying is repeated until a predetermined time interval has elapsed.

10. The method of claim 9 further comprising the step of registering a callback method in response to an expiry of said predetermined time interval.

11. The method of claim 10 wherein said steps of determining if a job is available for scheduling, determining if a session is available, and launching said session are performed in response to an invoking of said callback method by said target system.

12. A data processing system for connection scheduling within a network comprising a plurality of nodes, comprising:

circuitry operable for determining if a job is available for scheduling;

circuitry operable for determining, in response to said circuitry operable for determining if said job is available, if a session is available, wherein said session is included in a pool of sessions, said pool of sessions having a preselected one of a set of priority levels corresponding to a priority level of said job and wherein said session effects an execution of said job;

circuitry operable for creating a network connection to a target system for said execution of said job, wherein said target system is another node of the networked data processing system;

circuitry operable for launching said session to effect said execution of said job, if said session is available; and

circuitry operable for launching an error handling thread in response to an error condition, said error handling thread releasing said session.

13. The system of claim 12 wherein said session comprises a thread.

15. The system of claim 12 further comprising circuitry operable for determining if said network connection is an existing network connection, and wherein said circuitry operable for creating said network connection is operated if said network connection is not an existing network connection, and wherein said session is launched using said existing network connection if said network connection is an existing network connection such that said existing network connection supports multiple logical sessions.

17. The system of claim 12 further comprising circuitry operable for changing value of a job state from a first value to a second value in response to said launching of said error handling thread.

18. The system of claim 17 wherein said first value signals that said job is available for scheduling.

19. The system of claim 12 further comprising circuitry operable for retrying said steps of determining if a job is available for scheduling, determining if a session is available, and launching said session, in response to an error condition.

20. The system of claim 19 wherein said circuitry operable for retrying is operated until a predetermined time interval has elapsed.

21. The system of claim 20 further comprising circuitry operable for registering a callback method in response to an expiry of said predetermined time interval.

22. The system of claim 21 wherein said circuitry operable for determining if a job is available for scheduling, determining if a session is available, and launching said session are operated in response to an invoking of said callback method by said target system.

23. A computer program product embodied in a machine readable storage medium, the program product for job scheduling comprising instructions for:

determining if a job is available for scheduling;

determining, in response to instructions for determining if said job is available, if a session is available, wherein said session is included in a pool of sessions, said pool of sessions having a preselected one of a set of priority levels corresponding to a priority level of said job and wherein said session effects an execution of said job;

creating a network connection to a target system for said execution of said job, wherein said target system is another node of the networked data processing system;

launching said session to effect said execution of said job, if said session is available; and

launching an error handling thread in response to an error condition, said error handling thread releasing said session.

24. The program product of claim 23 wherein said session comprises a thread.

26. The program product of claim 23 further comprising instructions for determining if said network connection is an existing network connection, and wherein said instructions for creating said network connection are performed if said connection is not an existing network connection, and wherein said session is launched using said existing network connection if said network connection is an existing network connection such that said existing network connection supports multiple logical sessions.

28. The program product of claim 23 further comprising instructions for changing value of a job state from a first value to a second value in response to said launching of said error handling thread.

29. The program product of claim 28 wherein said first value signals that said job is available for scheduling.

30. The program product of claim 29 further comprising programming for retrying said steps of determining if a job is available for scheduling, determining if a session is available, and launching said session, in response to an error condition.

31. The program product of claim 30 wherein said instructions for retrying are repeated until a predetermined time interval has elapsed.

32. The program product of claim 31 further comprising programming for registering a callback method in response to an expiry of said predetermined time interval.

33. The program product of claim 32 wherein said instructions for determining if a job is available for scheduling, determining if a session is available, and launching said session are executed in response to an invoking of said callback method by said target system.

EVIDENCE APPENDIX

There is no evidence to be presented.

RELATED PROCEEDINGS APPENDIX

There are no related proceedings.